

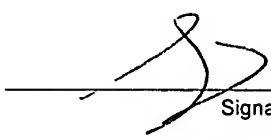


Doc Code: AP.PRE.REQ

PTO/SB/33 (07/05)

Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW		Docket Number (Optional)	
<p>I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR</p> <p>on <u>December 8, 2006</u></p> <p>Signature <u>Elaine F. Mian</u></p> <p>Typed or printed name <u>Elaine F. Mian</u></p>		Application Number	Filed
		10/019,330	3/7/2002
		First Named Inventor	
		Mihaly Toth	
		Art Unit	Examiner
		2143	Joseph E. Avellino
<p>Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.</p> <p>This request is being filed with a notice of appeal.</p> <p>The review is requested for the reason(s) stated on the attached sheet(s). Note: No more than five (5) pages may be provided.</p>			
<p>I am the</p> <p><input type="checkbox"/> applicant/inventor.</p> <p><input type="checkbox"/> assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)</p> <p><input checked="" type="checkbox"/> attorney or agent of record. Registration number <u>41,180</u></p> <p><input type="checkbox"/> attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34 _____</p>		<p><u></u> Signature</p> <p><u>Robert J. Mauri</u> Typed or printed name</p> <p><u>(203) 925-9400</u> Telephone number</p> <p><u>December 8, 2006</u> Date</p>	
<p>NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.</p>			

<input type="checkbox"/> *Total of _____ forms are submitted.

This collection of information is required by 35 U.S.C. 132. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11, 1.14 and 41.6. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



IN THE U.S. PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of:

APPLICANT: Toth et al.

SERIAL NO.: 10/019,330

FILING DATE:

March 7, 2002

EXAMINER: Joseph E. Avellino

ART UNIT:

2143

ATTORNEY'S DOCKET NO.:

44201757 PAR

TITLE: Server-Terminal(s) Session Management Involving Assigning Groups of Sessions to Threads

Mail Stop AF

Commissioner for Patents

P.O. BOX 1450

Alexandria, VA 22313-1450

PRE-APPEAL BRIEF REQUEST FOR REVIEW ATTACHMENT

The following is a concise recitation of clear errors in the Examiner's rejections in this application. Claims 1-21 are pending. All claims stand rejected under 35 U.S.C. §103(a) over a combination of Bayeh (U.S. Patent No. 6,098,093) and Freund et al., U.S. Patent No. 5,925,098. Claims 1, 18, 21, and 22 are independent claims, and each of these independent claims recites similar subject matter. Claim 1 is chosen for the following argument as being representative. Claim 1 recites the following:

A method of managing a plurality of sessions, the sessions being between a plurality of terminals and a server having a plurality of threads, the method comprising:

grouping the sessions into a plurality of groups; and

assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions.

It is respectfully submitted that none of the cited art includes the unique features of independent claim 1 and in particular the subject matter of "assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions." It is noted that a thread is assigned to each group of *sessions*, which means that the groups correspond to multiple sessions in the subject matter of "assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions".

Bayeh is directed to spreading requests among a number of servlets/web servers. In Bayeh, the requests are passed through a load balancing host 59, which sends the requests to web servers 60, 62, and 64. The load balancing host 59 sends requests to a “server selected according to policies implemented in the load-balancing host software.” Bayeh, col. 8, lines 42-58. It is believed that the “policies” are based on load of the web server and requests are sent to a web server based on load. The load balancing host 59 is not disclosed or implied as being one that would “group” the requests based on session. In fact, Bayeh appears to disclose that the load balancing host 59 acts only on requests and it is immaterial for purposes of balancing load as to which session it is that a request is related.

Because requests are spread among a number of servlets such that any servlet can handle requests from any session, more than one servlet might be able to access — at the same time — session information for a particular session. Bayeh discloses techniques for ensuring that only one servlet can access session information at any time for a particular session while requests can still be directed to any servlet. See, e.g., FIGS. 3, 4A, and 4B of Bayeh, and in particular steps 410-480.

In Bayeh therefore, there is no concerted effort or implication of grouping sessions into groups and assigning a thread to each group of sessions. Consequently, there is no disclosure or implication in Bayeh of “grouping the sessions into a plurality of groups” or “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions” as recited in claim 1.

Freund also does not disclose or imply the recited subject matter from independent claim 1. Instead, what Freund appears to disclose is a system for ensuring that all related requests (e.g., related through a specific transaction) are sent to the same thread. See, e.g., the following section of Freund:

A first embodiment of the server architecture (FIG. 2) involves the placing of a group 21 of FIFO queues 21a-21n with one request queue assigned to each execution thread 22a-22n in a one-to-one relationship. According to this embodiment, when client requests are received by the server's Object Adapter 23 over the Object Request Broker 24 from a client computer system, the Object Adapter 23 examines the contents of each request contained on its received request FIFO buffer 23a. Based on such contents the requests can then be forwarded on to the appropriate request queue 21a-21n. For example, if a first received client request relates to a particular transaction

and a second received client request relates to a different transaction, the first request can be assigned to queue 21a (and its corresponding execution thread 22a) and the second request can be assigned to queue 21b (and its corresponding execution thread 22b). Then, if a third received transaction request relates to the same transaction as the first request, the object adapter 23 would recognize this and assign this third request to the queue 21a to be processed by execution thread 22a.

In this way, *a complete transaction* consisting of many separate (but related) requests can be executed *by the same execution thread*, thus providing the same execution environment for each transactionally related request.

Freund col. 5, lines 3-27 (emphasis added).

Freund describes a transaction as the following: “A transaction defines a single unit of work that must either be fully completed or fully purged without action”. Freund, col. 3, lines 11-12. Freund also states the following: “According to these various embodiments, a scheduling mechanism ... ensures that all requests that are related (e.g. part of the same transaction) are sent to the same execution thread for processing.” Freund, col. 6, lines 39-43. The Examiner’s arguments imply that a “transaction” in Freund is equivalent to a “session” in Applicants’ claims (which Applicants do not admit).

There is no teaching or implication in Freund that multiple “transactions” are assigned to a single thread. In fact, it appears in Freund that a single thread is assigned to a single transaction:

According to these various embodiments, a scheduling mechanism (which does not necessarily have to be located in the Object Adapter) *ensures that all requests that are related* (e.g. part of the same transaction) *are sent to the same execution thread for processing*. This ensures consistency during the processing of an entire set of related requests. That is, the client machine issuing a sequence of transactionally related requests of the server machine can expect to get the same answer back when it issues the same sequence of requests at a later time. The processing conditions of the server's execution environment will stay the same because of the scheduling mechanism. That is, *intermediate requests belonging to another transaction (or not related to a transaction at all) are not allowed to be processed by the execution thread currently processing a transaction*. If such intermediate requests were allowed to be processed on a transaction's execution thread, the execution environment would be different when later parts of the transaction are processed by the thread and consistent results to report back to the client would not be guaranteed.

In order to determine whether a request belongs to a transaction, and the specifics of the transaction if it does, the Object Request Broker (ORB) 24 interrogates the transaction context of each incoming request. The transaction context of a request is obtained by the ORB by using the OMG-established Object Transaction Service (OTS) [OMG document 94.8.4 published in 1994]. The ORB also interrogates the Object Reference and any Service Contexts of the request to determine the specific server object (and thus server application) which the request is wishing to invoke. Once the transaction context and server context/application are determined, the ORB sends the request to the appropriate Object Adapter's queue. From there, the scheduling mechanism, as described in the above embodiments, ensures that all transactionally related requests are sent to the same execution thread. Also, *the scheduling mechanism can isolate the execution thread for a particular transaction by not allowing requests unrelated to that transaction from being processed on the transaction's assigned execution thread.*

Freund, col. 6, line 39 to col. 7, line 10 (emphases added). Consequently, Freund discloses that a single thread is assigned to a single "transaction" and there is no disclosure or implication that transactions are grouped and that a thread is assigned to a group of transactions.

Therefore, Freund does not disclose or imply "grouping the sessions into a plurality of groups" or "assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions" as recited in claim 1.

Because neither Bayeh nor Freud alone discloses "grouping the sessions into a plurality of groups" or "assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions", the combination of Bayeh and Freund does not disclose this subject matter. Therefore, independent claim 1 is patentable. Other independent claims 18, 21, and 22 contain subject matter similar to the subject matter in claim 1 and are also patentable, as are dependent claims 2-17, 19, and 20.


It is noted in the Advisory Action dated 20 December 2006, the Examiner stated the following (emphasis added):

As to point (1) it is the client requests, not the transactions, that the Office is stating is divided amongst the threads. The requests are divided up amongst the various threads based on the transaction identifiers. These transaction identifiers can be used to group like requests together such that all requests that are related can be executed on the same thread. *In this sense multiple sessions (i.e. requests) belonging to the same transaction can be grouped together on a single thread.* The transaction can be construed as the

entire group of requests that are assigned to a specific thread. By this rationale, the rejection is maintained.

The Examiner is apparently, therefore, equating a "session" of the claims with a "request" in Freund. However, Applicants define a "session" as the following: "A session is a *series* of interactions between a terminal and a server having a well-defined beginning and end and involving agreed-upon characteristics" (emphasis added). Page 2, lines 19-21 of Applicants' specification. It is clear from Freund that a "request" in Freund does not meet the definition given above by Applicants. A "request" in Freund appears to be a singular event and not a series of interactions. See Freund at col. 2, lines 4-12, where a request is implicitly defined as a single entity. Freund does disclose a transaction, which involves a series of client requests: see, e.g., col. 3, lines 11-12 of Freund ("A transaction defines a single unit of work that must either be fully completed or fully purged without action"); and col. 3, lines 24-27 ("Such transactions involve one particular client computer (e.g. 10) communicating with one particular server computer (e.g. 20) over *a series of client requests* which are processed by the server"; emphasis added). Regardless, the Examiner is equating a "request" in Freund with a "session" of the claims, and it is clear that a "request" in Freund does not meet the definition given above. Consequently, claim 1 (and similarly independent claims 18, 21, and 22) is patentable over Bayeh, Freund, or their combination.

Respectfully submitted:



Robert J. Mauri
Reg. No.: 41,180

8 December 06
Date

Customer No.: 29683

HARRINGTON & SMITH, LLP
4 Research Drive
Shelton, CT 06484-6212
Telephone: (203)925-9400
Facsimile: (203)944-0245
email: rmauri@hspatent.com